Victorian 6502 User Group Newsletter

KAOS

For People Who Have Got Smart

HARDWARE: . . . . . DAVID ANEAR
SOFTWARE: . . . . . JEFF RAE
LANGUAGES: . . . . . TONY DURRANT
AMATEUR RADIO: ROD DRYSDALE VK3BYU
EDUCATION: . . . . . NOEL DOLLMAN
LIBRARY: . . . . . JANNENE
TAPE LIBRARY: . . . . . RON CORK
NEWSLETTER: . . . . . IAN EYLES
SYM: . . . . . BRIAN CAMPBELL
ATARI: . . . . . GERRY MCCAUGHEY
SECRETARY: . . . . . ROSEMARY EYLES

OSI SYM KIM AIM ATARI APPLE UK101 ORANGE

Registered by Australia Post
Publication No. VBG4212

ADDRESS ALL CORRESPONDENCE TO 10 FORBES ST., ESSENDON, VIC. 3040

Vol.2 No.8 May 1982

COMP-SOFT now has stocks of the TASAN Video board kit. See the article by Paul Dodd on page 8. When you are ordering the board please state which buffers, 8T28 or 8T26 are on your computer and whether you will be changing your keyboard to C4 or leaving it C1.

Because of the amount of material we are receiving, we are having to print 16 pages each month, to make it easier for you to find articles , starting from this issue, we will be putting an index on the front page. We try to keep a balance of topics in the newsletter each month, but because so much work is being done on disk systems at the moment, most of the article submitted cover this topic. This means that some of the material has to be held over to later issues, so if the article you slaved over hasn't appearred yet, be patient. We realise that those on cassettes have been neglected, but a ray of sunshine (from the Sunshine State) has arrived. Ed Richardson of OSUG - Queensland User Group - is combining his newsletter with ours, and as his newsletter is cassette orientated, KAOS members can look forward to some very interesting articles, especially as Ed will be including items from the previous 23 newsletters.

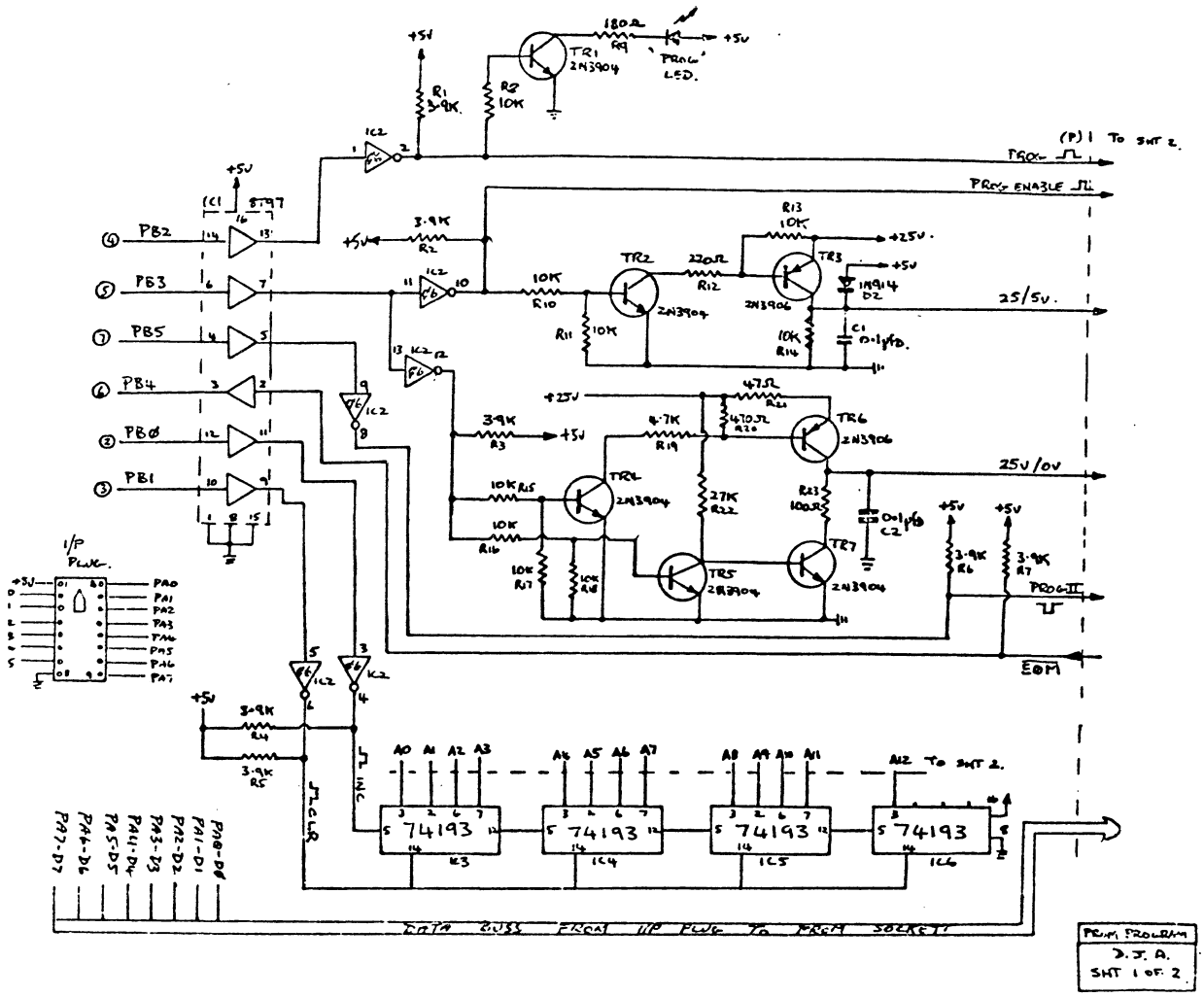
The next meeting will be held on Sunday 30th May at 2pm at the Essendon Primary School, corner of Raleigh St and Nicholson St, Essendon. The school children will be in as usual. The ATARI users now have enough members to form their own group and hold seperate meetings, venues etc. will be discussed at the meeting next Sunday, ATARI users please see note in M.A.C.E. column.

****

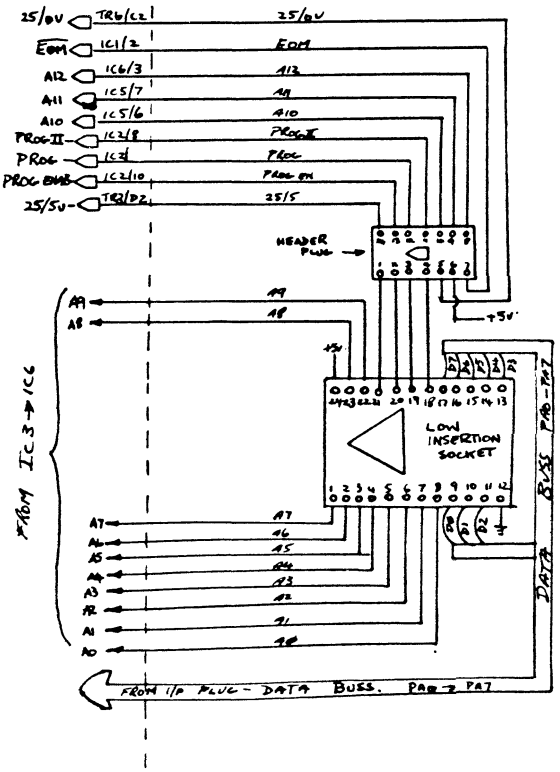
INDEX

EPROM PROGRAMMER CIRCUIT

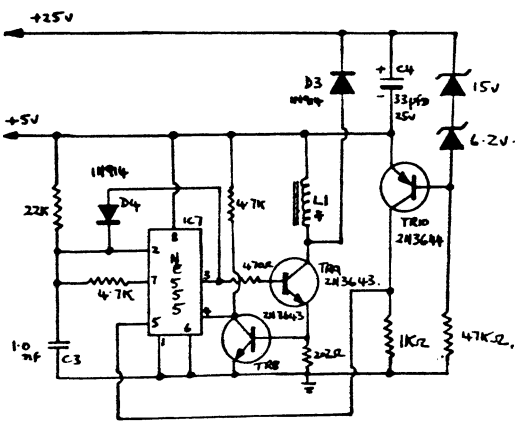
The circuits for David Anear's EPROM programmer which was described in Vol 2 No.7 newsletter.



FROM SHT 1.

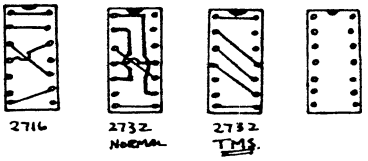


+25V SWITCHING SUPPLY FOR PROM PROGRAMMER.



L1 — 36→38 TURNS 16 GAUGE WIRE ON PHILIPS POT CORES.

HEADER PLUGS FOR VARIOUS PROMS.



PROM PROGS.
JDA SHT 2 OF 2

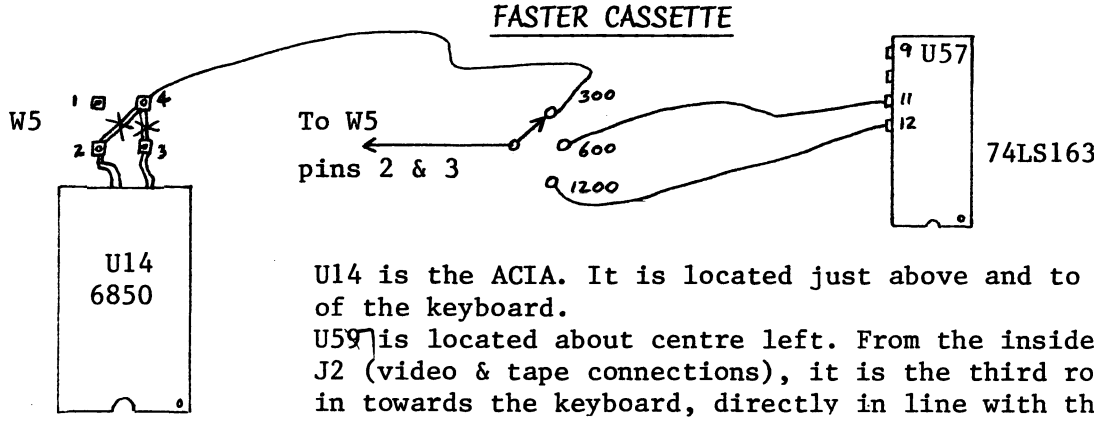
SUPERBOARD

Greetings from the Ohio Superboard User Group in Queensland. From July, 1982 we plan to present our newsletter as part of the KAOS Journal. KAOS will gain some money and a lot of new members. My members will gain by receiving all the extra information published by KAOS. I'll gain a rest from getting out a hundred letters each month. What's in it for KAOS members ?

Well, the quality of information we publish is second to none ! Our past 23 newsletters are packed with simple hardware mods, programs and useful routines and reviews of books and Commercial software. I plan to continue this when the newsletter is published as part of the KAOS Journal.

For the next couple of months, I'll be including some of the material from our past newsletters in this column and giving you some more details about us and our operation.

FASTER CASSETTE



U14 is the ACIA. It is located just above and to the left of the keyboard.
U59 is located about centre left. From the inside end of J2 (video & tape connections), it is the third row of ICs in towards the keyboard, directly in line with the end of J2. and the second IC to the right of the crystal.

Break the tracks at W5 between pins 2 & 4 and pins 3 & 4. Join 2 & 3 and connect to the moving arm of the switch as shown.

The faster save rate will work fine for a checksum, OSI load, or token format, but because BASIC loads each line into the input buffer and then tokenises and stores it, there are some problems for normal BASIC programs. Firstly, you will need a 2mHz clock for the 6502 at 1200 Baud. Secondly, to give BASIC the extra time it needs to tokenise and store the program, you will need to add some NULLS after the line feeds. When you SAVE, use the command NULL8:SAVE, or you can POKE the number of NULLS required into location 13. 2mHz and 600 Baud should work just fine without any changes to the NULLS.

A SIMPLE NON-DESTRUCTIVE MEMORY TEST FOR THE OSI

This program is located in the input buffer and can be loaded and run, testing all memory from $ØØ4Ø upwards, but leaving all programs intact. It will print in decimal, the first location which fails the test, for most machines, 8192

```
0014  A9 3F A0 00 85 11 84 12 E6 11 D0 02 E6 12 B1 11
0024  AA 38 A9 55 91 11 D1 11 D0 08 0A 90 F7 8A 91 11
0034  B0 E6 A5 12 A6 11 20 5E B9 4C 74 A2
```

The program will run on any OSI machine. It has been written so that it can be stored in ROM. By changing the first two opcodes to A9 01 and addresses 11 and 12 to 00 and 01, the routine would check from $ØØØ2 and up. Be sure to replace 4C and 74 in addresses $ØØØØ and $ØØØ1 before jumping to a Warmstart.

I'll be back next month with more,

Ed.

SUPERMON - SYSTEM RAM CONTENTS

The system RAM consists of 128 bytes addressed from $A600 to $A67F and exists in the 6532 RIOT (RAM,I/O and Timer). Due to partial decoding of the address bus, this device is addressed in a number of other areas in the memory map.

Between $A600 and $A61F is the scope buffer. This is where the text is stored for the oscilloscope display option. When you finally connect a terminal to your SYM, this area becomes free for your own use.

Next is the jump table between $A620 and $A62F. This is where the jump addresses are stored for the 'J' command. For example, at locations $A620 and $A621 is the start address for BASIC. When the command 'J 0' is entered, the program addressed by the above two locations is executed. This means that the user need not concern himself memorizing start addresses of programs. As an experiment for those with BASIC, try the following. Cold start BASIC with 'J 0'. In BASIC type -

Q=USR(&"8035",0)

You will now be back in the monitor. Put $7E in $A620 and $C2 in $A621. Type 'J 0' again and you will find that BASIC has warm started instead of cold starting as before.

The next 16 bytes are used as a scratchpad, although in V1.1 of the monitor some of these locations are assigned to the cassette I/O routines. This was demonstrated in Vol.2 No.5 of the KAOS newsletter.

The hex display buffer sits in the next 6 bytes; one byte per digit. The most significant bit corresponds to the decimal point. The rest control segments g, f, e, d, c, b, and a in that order. To use this facility, just store the display pattern required in these locations and call subroutine SCAND regularly, or subroutine HKEY once. The best way to learn this is to try it!

After three spare bytes comes PARNR at $A649. As described in last months newsletter, this location contains the number of parameters entered with a command. The parameters themselves follow in the next six bytes.

The next location is interesting. One of the problems with using old printers is that a carriage return can take some time. Since there is no buffer in these machines, a character sent too soon after a carriage return will either be lost or printed when the carriage is on its return journey. By increasing the value of PADBIT you can add a delay after every carriage return.

SDBYT controls the baud rate of the SYM. The values for the standard baud rates are given in the reference card, but there is nothing to stop you using any value you like.

ERCNT is obviously an erroe counter. It is used for counting errors in block move, fill, and load commands. If it is non-zero it is displayed at the end of the command.

TECHO allows echo and output suppressing. If you do not want to display what is being entered from the keyboard, clear bit 7. If you wish to suppress any displayed text, set bit 6.

TOUTFL masks the serial I/O devices. The reference card explains what setting each bit does.

KSHFL is a shift flag for the hex keyboard and is not important.

TV controls the delay generated by subroutine DELAY. This is used mainly by the debugger. A value of $0A generates a good speed for stepping through a program.

LSTCOM contains the last monitor command entered and is used in conjunction with PARNR and the parameters for command handling.

Since no-one is using paper tape for saving their programs, I will ignore MAXRC.

PCLR, PCHR, SR, FR, AR, XR, and YR are the user registers displayed with the 'R' command and can be modified before executing a user program.

INVEC and OUTVEC are the vectors to the terminal, hex keypad/seven segment display or oscilloscope I/O routines. If you have a printer or some other I/O device, just write the driver routine and patch it into the appropriate vector. A new cassette routine could be used this way, but there are better ways as you will see.

INSVEC detects the depression of a break key. The routine pointed to by this vector is dependant on the keyboard you are using. Mine is interrupt driven. (Every time a key is pressed, it is stored in a buffer. The input routine then reads the next key from the buffer). When the interrupt routine detects a control P, a flag is set. The routine pointed to by INSVEC reads and clears this flag. You may want to add a special switch. The SYM normally expects a prolonged START level on the CRT IN line. The BREAK key on the KTM-2 does this.

Those with V1.1 of the monitor will have URSVEC in the next three bytes. Those with V1.0 will have three spare bytes. URSVEC uses the next three bytes. The function of these was described last month.

SCNVEC points to the hex dispaly scanning routine and should be left alone.

EXEVEC and TRCVEC are not related but have special functions to be described in separate articles.

Since the IRQ pin and BRK instruction use the same interrupt vector, the monitor will sort out which is which and use UIRQVC or UBRKVC accordingly if the interrupt vector IRQVEC is left alone.

NMIVEC is used for the debugger and is tied closely with TRCVEC and will be discussed later.

RSTVEC is the reset vector but it is unused. If you remember the second article on the SYM, I said that it uses two locations at the top of a ROM. RSTVEC in the system RAM has no meaning at all.

Lastly, IRQVEC is obviously the interrupt vector. It normally points to the IRQ/BRK routine.

This is how SUPERMON uses the system RAM. If you were to write your own monitor you could use it any way you like. Experiment with the system RAM contents until you understand them. Do not forget to use subroutine ACCESS to remove the write protection. Have fun and be game - you cannot break anything! (Except your sanity, of course.)

NEXT MONTH - CASSETTE I/O ROUTINES AND SYSTEM RAM ADDRESSING

As you have noticed, we did not get them in as promised last month due to the length of this article. Next month for sure! Also we will look at how the SYM moved its IRQ and NMI vectors down into the system RAM. We will also see how to grab an extra 2K of addressing space for, say, a new reset routine ROM.

Brian Campbell

A BEXEC* WITH "MENU" AND "RUB OUT" KEY

FOR COMP-DOS 1.2

Here's a BEXEC* which includes a really useful feature - you can use the "RUB OUT" key for 'destructive' (ie. disappearing) back space! It's much easier than 'Shift O'.

This BEXEC* will also boot up with a numbered program 'menu' (also showing track numbers) which will then RUN the selected program number.

You will need to change line 60 to call the track with appropriate memory for your machine, ie. Track 16 for 32K and Track 17 for 24K (see COMP-DOS manual).

```
  1 REM    ## BEXEC*  (UPDATE MAY 1982) for DOS 1.2  ##
  2 REM
  4 REM  * DESTRUCTIVE BACK-SPACE USING 'RUB OUT' KEY
  5 POKE1386,128:POKE1394,127:POKE1419,127:POKE2820,127
  9 REM                          *  I/O FLAGS
 10 X=PEEK(10950):POKE8993,X:POKE8994,X
 20 HM=PEEK(8960): REM           *  MEMORY PAGE COUNT
 50 REM                          *  CALL TRACK WITH 40K DOS
 60 DISK!"CA 9A00=15,1":DISK!"GO 9A00
 65 HM=HM-6:POKE133,HM:POKE8960,HM:REM * END OF BASIC WORK SPACE
 70 POKE2073,173:    REM         *  ENABLES CONTROL C
 75 POKE2893,55:     REM         *  BREAK ON C/R
 80 POKE2894,8:      REM         *  "READO FROM START"
 85 POKE2888,0:      REM         *  C/R ON 'INPUT' = "" OR 0
 95 CLEAR
100 DIMN$(64):       REM         *  DIMENSIONS No. OF ENTRIES
105 C$="CA 2E79=12,":REM         *  CALL DIR. TRACK TO RAM
110 P=11897
115 DEFFNA(X)=10*INT(X/16)+X-16*INT(X/16)
130 POKE8955,213:POKE8956,252:REM * USR(X) POINTER
135 X=USR(8)
140 PRINT"     NAME OF DISK":PRINT"-----------------
175 DISK!C$+"1":REM              *  CALL DIR. TRACK - SECTOR 1
180 GOSUB200
185 DISK!C$+"2":REM              *  CALL DIR. TRACK - SECTOR 2
190 GOSUB200
195 GOTO270
199 REM       *  READ DIRECTORY OUT OF BUFFERS INTO ARRAYS
200 FOR1=PTOP+248STEP8
205 IFPEEK(1)=35THEN260:REM      *  LOOKS FOR EMPTY (#) ENTRY SPACE
210 A$=STR$(N+1)
215 M$=MID$(A$,2)
220 N$=""
225 FORX=ITOI+5
230 N$=N$+CHR$(PEEK(X))
235 NEXTX
240 N$(N)=N$
245 N=N+1
250 REM                          *  PRINT OUT DIRECTORY
255 PRINTM$TAB(3)N$TAB(12)FNA(PEEK(I+6)TAB(16)"-"TAB(17)FNA(PEEK(I+7))
260 NEXTI
265 RETURN
270 PRINT
275 PRINT"----------------------------
280 INPUT"  YOUR SELECTION";X:PRINT
285 IFX=0THENNEW
290 IFX=1ORX=4THENPRINT"YOU CAN'T RUN 'OS65D3' OR 'DOS1.2'":GOTO280
295 IFX>VAL(M$)THENPRINT"THERE ARE ONLY "M$" PROGRAMS":GOTO280
300 POKE2888,27:  REM            *  DISABLES C/R ON 'INPUT' = "" OR 0
305 RUNN$(X-1)                                                  Bill Roberts
```

NEW PROMPT FOR BASIC INPUT ON SUPERBOARD

Some BASICs allow you to define the character used as a prompt for an INPUT statement. OSI BASIC doesn't. However, there is a very easy way to do it using a small M/C routine. In locations $021A,$021B is a vector to the screen print routine, which prints the character in A (the 6502 accumulator). We change the vector to point to our routine, and when we are finished we jump to the screen print routine so BASIC doesn't know we have interferred.

The vector in 24X24 screen format is $FF69. If you have DABUG, and are using 12X48, the vector is $FB00.

The routine in M/C;

```
021A 35                 ;vector changed to
021B 02                 ;point to our routine

0235 C9 3F              CMP #'?
0237 D0 02              BNE FIN
0239 A9 3F              LDA #'?
023B 4C 69 FF FIN       JMP #FF69
```

This routine, as in the listing, does nothing - it substitutes a '?' for a '?'. However, POKE 570,X (where X is a non-graphic ASCII character) will change the prompt to X. It also changes every other '?' to X, but except in some PRINT statements, this will not matter. Should you want to print a '?' you only need to POKE 570,63. This makes the routine replace a '?' with a '?' as in the listing.
NOTE:- 570=$023A

As this routine is useful only in BASIC, here is a BASIC program to load it.

```
10 FORR=1TO9:READA:POKE564+R,A:NEXT:POKE538,53:POKE539:,2
20 DATA201,63,208,2,169,63,76,105,255
```

The last two bytes in data line 20 ie. 105,255 should read 0,251 for use with 12X48 screen. It may be a good idea to poke the vector back to normal at the end of the program -
eg. 9999 POKE538,105:POKE539,255
or for 12X48
999 POKE538,0:POKE539,251

The program sits in spare bytes below the start of BASIC, so it is safe from being overwritten while the program is running (unless you POKE at it accidentally).

The idea for this program came from an O.S.U.G. newsletter (Brisbane User's Group) where a call to a M/C routine is made when a special character is printed. This idea can be extended for multiple M/C calls using say a '#' as the special character, followed by a letter or letters naming the routine. The name can be obtained by the M/C routine by incrementing the pointer BASIC keeps to indicate the current position along the line. This is a convenient way to extend the BASIC instruction set.

Darryl Lock

THE TASAN VIDEO BOARD

COMP-SOFT have had the C1/C4 Video board for a couple of weeks now, and finally all of the bugs have been ironed out (I hope!).

In answer to a few of your questions, the Video board will give you 64 characters in 32 rows (64X32 format) in the C4 pattern, 32X32 in either the C4 or the C1 format, however to get the C1 format display, you must short a point on the Video board. to ground after switching to C4 32X32 mode. If you don't want the C4 32X32 format (which most of you won't, since it's pretty useless anyway!) you can connect the pad on the board marked "32X32" to the track immediately adjacent to it, (there is only one track nearby) and then, whenever you switch to 32X32 format, it will be in the C1P 32X32. display.

One trouble with the board is that if you have enabled the colour RAM, or switched to 32X32 characters mode, and you hit "BREAK" the background may be filled with annoying inverse video blocks or you may not be able to see the "D/C/W/M?" prompt. The cure for this is very simple: on the Superboard, connect U8 (the 6502 chip) pin 40 (the RESET line) to J1 pin 11 (the unused pin on the 40 way connector). This "mod" is compatible with the Tasker Buss. Then on the Video board connect either J2 or J3 (either of the 40 pin sockets) pin 11 to U 51 pin 1. Whenever you hit "BREAK" now, it will turn the colour RAM off and switch you to 64 character format.

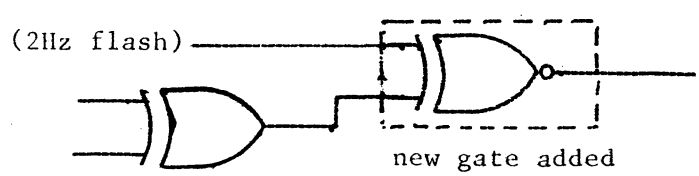
The Video board has 4 bits of colour RAM for every character position, David Tasker has used one of these bits for inverse video (bit 0), and the other three are supposedly used to select one of the eight colours. You can only get actual colour by adding an extra board and as most of us do not want to go to the trouble or expense of buying and building a colour board, or chopping up the family colout tv to make a video monitor (Yes I know you can use RF but.....), these other bits can be used for more immediate purposes.

I chose to use bit 1 (the second of the colour bits) to generate double brightness characters for highlighting. This "mode" is very simple. Just connect a resistorof around 30K (very roughly) depending on how bright you want your "double bright" characters, from U22 pin 7 to the video out pad, underneath the board.

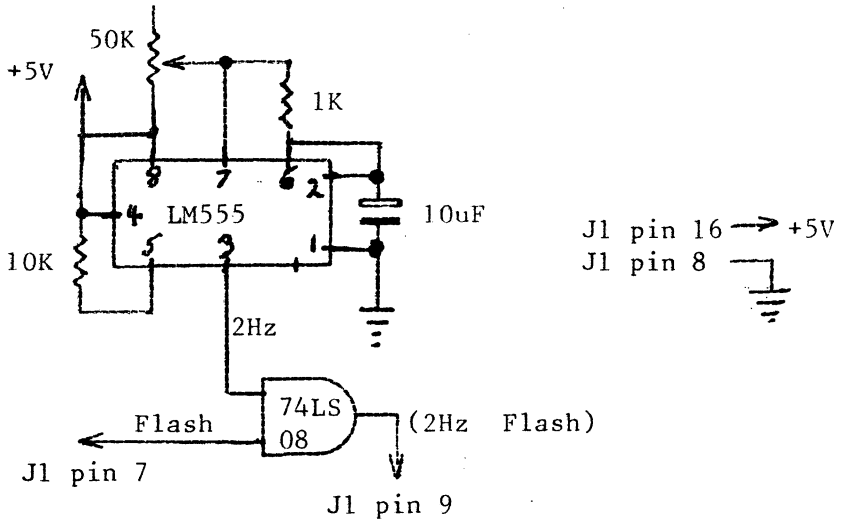
I chose bit 2 to be the flashing character bit. My version of flashing characters alternates the character with its inverse at a rate of about 2Hz (this is ajustable to any rate you like). You need to do a few modifications to the Video board:-

Cut the track from U36 pin 12 to U37 pin 11
Join U36 pin 12 to U37 pin 6
Join U37 pin 5 to U37 pin 11
Join U37 pin 4 to J5 pin 9 (that's the spare socket on the right of the board)

The new video circuit looks something like this:



Now we need to generate the (2Hz Flash) signal. This is done on a little external piece of vero-board which plugs into J5, the colour board socket, via a 16-pin wire wrap socket. On the vero-board (my piece was about 4cm square) build the following circuit:



J1 is the 16 pin wire wrap socket.

NOTE:
Pin 16 is connected to +5v on the Video board
Pin 8 is connected to ground on the Video board

This circuit probably could be made using just a 74LS00, using two of the gates to make a 2Hz oscillator, a third to NAND this signal with the 'flash' signal and the forth to convert the NAND back to an AND.

I am considering using bit 3 to generate programmable characters, but more on that next month.

Before anybody goes rushing ahead and reversing the diodes on the Superboard keyboard, there are probably other ways to generate a C4 keyboard. One method would be to use a location in Page Zero memory to tell the computer whether it had a normal (C1P) keyboard or reversed (C4P) keyboard, and have the new DABUG doing the switching under software control. Another method would be to use the two 14 pin sockets on the Video board (which are for an external keyboard), and connect the Superboard keyboard to this via a set of 8T28s or 8T26s (non-inverting and inverting buffers) and have one bit on the keyboard latch determine which set of buffers to use. Anyway happy Video board making, and thankyou David Tasker and David Anear for the board.

Paul Dodd

****

LIBRARY NEWS

The library is progressing well, with many magazines being borrowed. Unfortunately, many of these magazines are not being returned. If you cannot return a magazine for any reason, please contact me. If I am not contacted there is a fine for late returns to the library.

The library has a complete set of Australian Personal Computer up to March 1982 - with the exception of Vol.1, Issue 1. - Any offers??? If you wish to read the latest APC, KAOS Library is the place to see it.

Would the following KAOS members contact me:
W. Babb C. Baker L. Cresswell R. Northmore

Jannene

REPORT OF THE 1982 WEST COAST COMPUTER FAIR

Any arcade game now available on the other computers and in the arcades is now, or will soon be, ready for the ATARI. Several of the large APPLE vendors say that with Apple sales off and Atari sales up, they are proceeding as quickly as possible to enter the Atari market full force, so come on Future-Tronics, how about giving Atari owners a fair go and start selling some decent software.

Just released is the new double density drive from Percom. It sells for $799U.S. for one and $1200U.S. for two. Just think, 4 drive capacity for only $A1060, it is fully compatible with the Atari 810tm* disk and uses the serial port.

Want to speed up your Basic programs? Well now you can withthe Fast Chip. A $42.00 chip that replaces the O.S. Floating Point ROM and speeds up Basic programs dramatically, from Newell Ind, 3340 Nottingham In, plano, TX 75074.

An EPROM programmer, and a Trac Ball have been released by Roklan Corp. Broderbund Software released an excellent new arcade game called Apple Panic. It has similarities to Donkey Kong. Coming soon, Midnight Magic a pinball game, Trac Attach an arcade game, and Star Blazer a space game. They are all popular Apple games.

Datasoft have released 5 new products for the Atari. Micro Painter, allows you to draw pictures on your TV screen. Very good graphics. Bishop's Square allows the user to load in a picture from Micro Painter and then juggles it into a jigsaw puzzle for the user to do. Tumble Bugs is an outstanding variation on Pacman. Graphics Master is another graphics maker program that allows you to withdraw parts of one picture and put them on another. Last of all is a graphics generator program for making graphic characters, just press a single key, press C it will draw a car, press H it will draw a house, you can then insert them into your own program.

On Line Systems have released/announced 5 new programs for the Atari. Threshold is the best arcade game for the Atari ever released. Cross fire, a strange but well done new type of arcade game. The Guns of Neburon is expected to sell thousands of Atari computers on its own, we will just have to wait and see. Frogger has been completed and it is superb. Time Zone is an adventure game to end all adventure games, it is a full 20 disk adventure. There are 8 time zones and 8 continents in each time period, including interplanetary, to explore, all in hi-res graphics.

Gerry Mc Caughey

WHAT'S NEW IN AUSTRALIA

Great news for all you Atari owners, Futuretronics will be importing programs from the A.P.X. collection. These programs range from Utilities, Home Finance, Education and of course some great video games. Atari created the APX (Atari Program Exchange) to distribute user-written software for Atari Computer owners.

A new book from the Wiley Publishing Co. has been released for the Atari 400/800. The book is called 'Atari Graphics and Sound' and will be available through M.A.C.E. for $13.95 plus $2.00 for postage. Orders within 30 days please.

IMPORTANT NOTICE

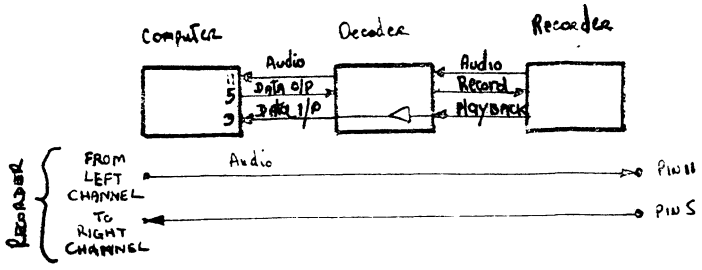
Will all M.A.C.E. members please try to attend the next meeting as we will be discussing a breakaway from KAOS to become a users group for Atari onlv. Those members who cannot attend please ring or write to Karl

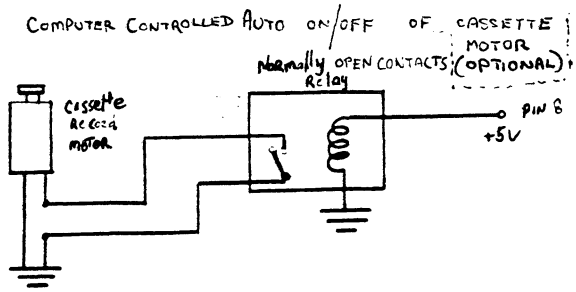
Also will all paid up members please notify Karl of their name and address. Our filing system was mislaid and we no longer have a reference to all our members.

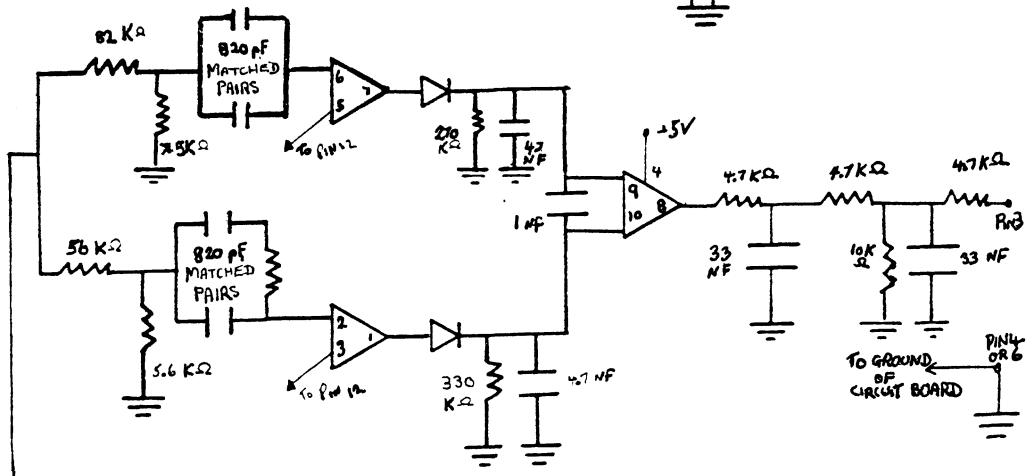
Karl Valentic

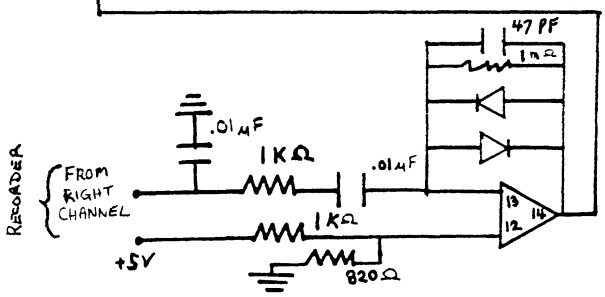
CASSETTE RECORDER INTERFACE FOR THE ATARI 400/800 COMPUTER

by Denis Biddle

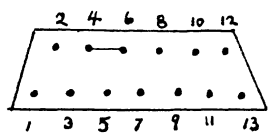








The pins to use on the Serial I/O Jack



4+6 ground

8 Motor control of cassette and +5 volts for external

5 Data output, connect directly to record in of cassette, Adjust record level for no distortion

3 Data input, connect to the decoder

11 Audio input, the left channel of you recorder

Resistors (Ohms)

1 X 820	2 X 15K
2 X 1K	1 X 56K
3 X 4.7K	1 X 82K
1 X 5.6K	1 X 240K
1 X 7.5K	1 X 270K
1 X 10K	2 X 330K

Misc.

1 X IC LM324N Quad OP AMP
4 X Silicon Diodes, Low power

Capacitors (Disk Ceramic)

4 X 820pF
1 X 47pF
1 X 1nF
2 X 4.7nF
2 X 33nF
2 X .01uF

Other

1 X 5volt relay
1 X jiffy box

NOTE

Do not connect the ground of the computer to the ground of the recorder
Use shielded wire to interconnect between computer and recorder
There are 5 wires going to the cassette recorder, 2 for motor, 1 for audio, 1 for right channel record, and 1 for replay

THE MEETING WAS KAOS

After being absent for a number of months David Anear had the pleasure of welcoming Professor Luxenberg to the April meeting of KAOS. Without hesitation Prof. Luxenberg proceeded to give a most informative talk on how he first became involved in computers, in particular the Kim and Sym. He expressed to the meeting how important he felt it is to start at the very beginning with the most basic computer which would only support machine code programming and a simple monitor. It was an interesting and most humorous discussion which reflected his early struggles and successes with the Kim and Sym. Prof.Luxenberg also indicated to the meeting that he was surprised there was so much activity in electronics in Australia, and that the meeting was the largest number of 6502 users he had seen assembled in one place. After the meeting Prof. Luxenberg rallied with the Kim and Sym users and I imagine that quite a lot of useful information was exchanged.

EXTENDED BASIC:- The Macauley brothers used their computer the "Mirror" to demonstrate the possiblity of a standardization for an "EXTENDED BASIC" which could include a number of graphic commands as demonstrated in the Mirror. The subroutines would be transparent to the programmer, therefore no knowledge of machine language would be necessary to use the extended basic. There was considerable discussion on the types of commands that could be included, this resulted in a number of members joining a group to consider the whole concept.

FORTH:- David Wilson addressed the meeting and indicated that he was considering forming a FORTH interest group within KAOS. David asked those members who may be interested to speak to him after the main meeting. Country and Interstate members can contact David through KAOS.

E'PROM PROGRAMMER:- John Whitehead has written a Basic program to drive an E'Prom programmer and has made it available to KAOS members. I believe it is based on David Anear's programmer. Contact John for more information

TRASH & TREASURE:- It was announced during the meeting that the JUNE meeting would include an afternoon of Trash & Treasure. So bring along all those bits and pieces you never know what to do with. You may be able to finance your next project.

QUESTIONS:- A number of questions were asked during the meeting such as "What is RS232?", these were answered by the hardware committee. Members are encouraged to ask questions during the meeting, or preferably, write the question on the pad available on Rosemary's table. If those people who can't get to meetings write to us the question will be answered at the meeting (no names) and also answered by letter. No matter how basic they appear, your questions are always of interest to someone else, I know I certainly gained new information from listening to the answers.

BIG TRACK:- David Anear caused quite a stir with his son's ??? new toy called "BIG TRACK". It is an example of the new generation of microprocessor controlled toys from the U.S.A. Hopefully David might review BIG TRACK in the near future.

73s from MELBOURNE
ROD DRYSDALE

****

FOR SALE

ClP in case, programs and documentation, Tasker Buss and Power supply.

$750.00 ono Godfrey Van Der Linden

Godfrey will be in Melbourne for the May meeting, so contact him there, or contact KAOS.

CORRECTIONS, MODIFICATIONS & IMPROVEMENTS TO DISK BASIC

The standard Disk Basic supplied with OS65D Version 3.2 does not use the ROR instruction. This is because early 6502 micro-processors (pre June 1976) did not have this instruction. However, BASIC-in-ROM does use the ROR instruction, so if you can run BASIC-in-ROM, then there is no reason why your Disk BASIC cannot make use of this instruction.

My modified BASIC runs scientific type programs up to 40% faster. Consider these times for the standard benchmark program eight (100 square, log and sine operations):

Standard 9 digit Disk Basic	17.5 seconds
Modified 9 digit Disk Basic	11.0 seconds
6½ digit Basic in ROM	7.5 seconds

Non-Scientific programs typically run 5 to 10% faster. Apart from the reduced running times, the major advantage of making use of the ROR instruction is that it frees up over 100 bytes which can be used for adding extra functions to BASIC.

In general the form of the modification is that the sequence of instructions:-

```
        LDA #00
        BCC SKIP
        LDA #80
SKIP:   LSR x
        ORA x
        STA x
```

is replaced by the single instruction:-

```
        ROR x
```

converting 12 bytes to 2 (typically).

The improved code is as follows:

```
17A5 90 0E     BCC 17B5
17AB 66 AF     ROR AF
17AD 66 B0     ROR B0
17AF 66 B1     ROR B1
17B1 66 B2     ROR B2
17B3 66 BD     ROR BD
17B5 60        RTS
184A 2A        ROL A
184B 76 01     ROR 01,X
184D 4C 54 18  JMP 1854
```

```
1854 76 02  ROR 02,X
1856 76 03  ROR 03,X
1858 76 04  ROR 04,X
185A 68     PLA
185B 08     PHP
185C 6A     ROR A
185D 28     PLP
185E C8     INY
185F D0 E6  BNE 1847
1861 18     CLC
1862 60     RTS
```

```
1946 66 73  ROR 73
1948 66 74  ROR 74
194A 66 75  ROR 75
194C 66 76  ROR 76
194E 66 BD  ROR BD
1950 98     TYA
1951 4A     LSR A
1952 D0 96  BNE 192A
1954 60     RTS
```

These modifications leave 17B6 to 17E7, 1862 to 1884 and 1955 to 1986 available for extensions to BASIC. First however, there is at least one error in DISK BASIC (which may have been corrected in recent versions). Direct Access data files longer than 16 tracks (256 records with mini-floppies) do not work correctly. This will cause problems with Data-Base applications which might attempt to use all or most of a disk for a large data file. The error lies in track 12, sector 4 (which is brought into 2E79 to 2F78 by a DISK OPEN operation). A binary track offset is added to a packed decimal base address using decimal mode. This works for track offsets from $00 to $0F, but $10 is interpreted as 10 (decimal) instead of 16 (decimal).

The fix is as follows:-

```
2F18 20 6F 19  JSR 196F  (or wherever you want to put the subroutine)
196F AD 92 2F  LDA 2F92
1972 38        SEC
1973 A2 FF     LDX #FF
1975 E8        INX       compute the number of tens
1976 E9 0A     SBC #0A
1978 B0 FB     BCS 1975
197A 69 0A     ADC #0A
197C 85 FA     STA FA    save the remainder
197E 8A        TXA       move the count of tens to the high nibble
197F 0A        ASL A
1980 0A        ASL A
1981 0A        ASL A
1982 0A        ASL A
1983 05 FA     ORA FA
1985 60        RTS
```

This merely converts the binary offset to a packed decimal offset and takes advantage of some of the space we have created within BASIC.

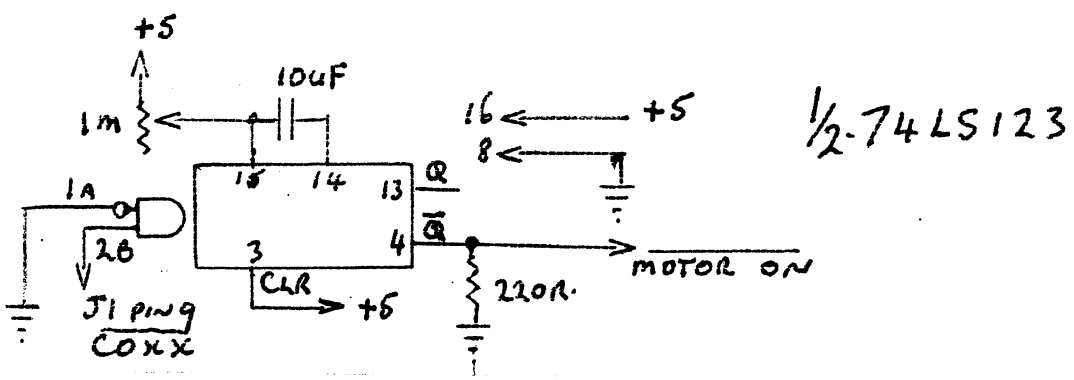
Next month I shall describe how I have used the remaining space. There were a number of typing errors in last months article, *Not mine! (Rosemary)* mostly missing # characters. Note also that some of the constants are not actually used in the sample program. Also DISKO means DISK OPEN (only the first character is needed). The most serious error was line 2000 which should read 2000 PRINT#5,B$;A$; This statement adds a line to the command file.

Rodney Eisfelder

****

DISK DRIVE SWITCH #2

This version of the disk drive motor control switch will automatically switch the drive motor on when you do a disk access, and then switch the motor off again about 10 or 15 seconds after the last disk access.



The $\overline{\text{MOTOR ON}}$ signal goes to pin 16 on the drives edge connector. +5V can come from pin 22 (or 23 or 24) on the molex connector on the Tasker floppy board. Ground can come from pin 1 (the ready drive 1 signal) or pin 18 or pin 20 on the molex connector. The $\overline{\text{COXX}}$ signal comes from 8 of IC8 on the Tasker board (that's the 74LS05 chip).

A neat way of building this circuit is to incorporate it onto your floppy disk connector cable, on the veroboard that converts fromthe molex connector to the 34 way cable (if you use one of these). I cut the track on the Tasker board, to pin 9 of the molex connector, which isn't used with B51 or B52 drives, and connected the $\overline{\text{COXX}}$ (pin 8 of IC8) to this pin.

Paul Dodd

USEFUL DATE CONVERSION ROUTINES

Recently I had occasion to use some date conversion routines in a program I was writing at work and it occured to me that some other KAOS readers may find them useful.

Some of you may be asking "What are date conversion routines and why do we need them ?". Date conversion routines allow us to convert between the various date formats that are used in data processing, namely:
(i) Gregorian date (ii) Julian date (iii) Serial date

I will explain what each of these are, but first I would like to head off any comments from purists who may wish to contest my definitions by saying that the definitions I use are those that are commonly used in commercial data processing.

GREGORIAN DATE-

This is the date as it is most usually expressed in everday use namely, day/month/year or as it is often written by programmers, DD/MM/YY. As an example, I am writing this article on Saturday the 22nd of May 1982 and this date would be expressed in Gregorian form as 22/5/82.
Note - we usually leave out the first 2 digits of the year i.e. 1982 is usually written as 82.

Julian Date

- This is the second most common way of expressing the date in data processing but is only rarely used in everyday life with perhaps the exception of calenders used in offices of the "bunch of dates" variety. In this format the date is written as YYDDD where YY is the year and DDD is the serial day number within the year. Using the above date as an example we would write the Julian date as 82142 i.e. the date 22/5/82 is the 142nd day of 1982.

Serial Date

- This date format is practically never used in everyday life and it is not used as often as it should be in commercial data processing. In this format the date is expressed as a single number usually of five digits. Using the example above the Serial date would be written as 30033 if day 1 was 1/3/1900 or as 30093 if day 1 was 1/1/1900 i.e. this article was written on the 30033rd day using 1st March 1900 as day 1

The BASIC routines in the program below allow for conversion between the various formats without the need for tables of constants within the program. The formulae are valid for the date range 1st March 1900 to 31st December 1999 inclusive however the Gregorian to Julian and Julian to Gregorian formulae may be easily modified to be valid from 1583 onwards.
The routines allow for the following conversions:-

(i) Gregorian to Serial (line numbers 1000-1030)
(ii) Serial to Gregorian (line numbers 1100-1155)
(iii) Julian to Gregorian (line numbers 1200-1250)
(iv) Gregorian to Julian (line numbers 1300-1345)

The two other possible conversions of Julian to Serial and Serial to Julian must be done in two steps each.

And what possible uses can the average KAOS member put these routines to ??? The answer is up to each of you but a couple of uses which come to mind are:-

(a) Date validation - this can be performed by the simple expedient of converting a Gregorian date to Serial and then converting the Serial answer back to Gregorian and comparing the resulting Gregorian date to the original Gregorian date. They will only be identical if the original date was in fact valid. Try this for example on 29/2/82.
(b) Biorythm programs - you can try writing your own biorythm programs using the Gregorian to Serial conversion routine to make life easier for yourself.
(c) How old are you (in days)? - this is an easy computation if you convert both today's date and your birthdate to Serial form.

The simple driver program accompanying the four routines was used by me in checking them when I coded them in BASIC for the first time.
The only thing you should beware of is that you look at each routine in turn and find out the names of both the input and output fields. Also, do not assume that input fields remain uncorrupted after use unless you have determined this from the code of the routine(s) you are using.
By the way, I claim no originality in the algorithms used. They are all standard algorithms used in the industry and come from various sources.

```
10 REM TEST MAINLINE
20 INPUT"D,M,Y";DI,MI,YI
30 IFDI=99THEN STOP
40 GOSUB1000
50 PRINT"SERIAL DAY NO = ";KS
55 KI=KS
60 GOSUB1100
70 PRINT"D,M,Y = ";DO;",";MO;",";YO
75 DI=DO:MI=MO:YI=YO
80 GOSUB1300:PRINT"YYNNN = ";YI;",";JO
85 JI=JO:GOSUB1200
90 PRINT"D,M,Y = ";DO;",";MO;",";YI
95 GOTO20
1000 REM GREGORIAN TO SERIAL
1005 REM INPUT FIELDS ARE DI, MI, YI
1010 REM OUPUT FIELD IS KS
1015 IF MI)2 THENMI=MI-3:GOTO1025
1020 MI=MI+9:YI=YI-1
1025 KS=INT(1461*YI/4)+INT((153*MI+2)/5)+DI
1030 RETURN
1100 REM SERIAL TO GREGORIAN
1105 REM INPUT FIELD IS KI
1110 REM OUTPUT FIELDS ARE DO, MO, YO
1115 YO=INT((4*KI-1)/1461)
1120 DO=4*KI-1-1461*YO
1125 DO=INT((DO+4)/4)
1130 MO=INT((5*DO-3)/153)
1135 DO=5*DO-3-153*MO
1140 DO=INT((DO+5)/5)
1145 IF MO(10 THEN MO=MO+3:GOTO1155
1150 MO=MO-9:YO=YO+1
1155 RETURN
1200 REM JULIAN TO GREGORIAN
1205 REM INPUT FIELDS ARE YI, JI
1210 REM OUTPUT FIELDS ARE MO, DO
1212 TW=0
1215 IF INT(YI/4)*4=YI THEN TW=1
1220 IF JI)(59+TW)THEN WW=2-TW:GOTO1230
1225 WW=0
1230 DO=JI+WW
1235 MO=INT((DO+91)*100/3055)
1240 DO=DO+91-INT((MO*3055)/100)
1245 MO=MO-2
1250 RETURN
1300 REM GREGORIAN TO JULIAN
1305 REM INPUT FIELDS ARE DI, MI, YI
1310 REM OUTPUT FIELD IS JO
1315 W1=INT(3055*(MI+2)/100)-INT((MI+10)/13)*2-91
1320 W2=INT((YI-INT(YI/4)*4+3)/4)
1325 W2=1-W2
1335 W2=W2*INT((MI+10)/13)
1340 JO=W1+W2+DI
1345 RETURN
OK
```

Malcolm Coghill
22/5/82

****

FOR SALE

CHALLENGER 1P Series 2 in metal case. 8K RAM RF Modulator
Some software Approx. 1 year old $400.00 or offer

John Craig

COMPUTER UK101 (similar to OSI C1P) with new Monitor and revised BASIC 3 ROM chip. Complete with power supply, VHF modulator and case.
(48X16 screen) $325.00

phone Edward Smeda (evenings)

TELETYPE MODEL 38 15 in. TRACTOR/FRICTION FEED
Serial computer interface 110 baud generator and
negative voltage supply and box of paper. PRICE $480.00 ono

W.J. Clarkson

CORRECTIONS TO THE ARTICLE ON FORTH Vol.2 No.7 page 10

Lines should read:-

```
1. HEX 0    WARNING    !
2.     400      '     B/BUF
>R 80 - DUP DUP R> B/BUF 4 + *
6. HEX : BLOCK-INIT CLR
DO FIRST I DUP . CR OFFSET @ +
```